Preparación de Datos

1. Unificación de Bases de Datos

En el marco del reto, se identificaron un total de cuatro bases principales, que se integraron de forma estratégica para consolidar la información relevante:

- prueba_op_base_pivot_var_rpta_alt_enmascarado_trtest.csv: Contiene la variable objetivo para el conjunto de entrenamiento.
- prueba_op_probabilidad_oblig_base_hist_enmascarado_completa.csv: Incluye probabilidades generadas por modelos previos.
- prueba_op_master_customer_data_enmascarado_completa.csv: Proporciona información demográfica de los clientes.
- prueba_op_maestra_cuotas_pagos_mes_hist_enmascarado_completa.csv: Aporta datos históricos de pagos y obligaciones.

Como base maestra, se seleccionó prueba_op_base_pivot_var_rpta_alt_enmascarado_trtest, ya que contiene la variable objetivo. A partir de los campos NIT enmascarado y NIT de la obligación, se realizó el cruce para unificar las demás bases de datos.

Adicionalmente, el reto incluye una base específica denominada prueba_op_base_pivot_var_rpta_alt_enmascarado_oot, que contiene los clientes a los que se realizará la predicción. Esta base no se emplea en el entrenamiento del modelo, sino exclusivamente en la fase de inferencia.

2. Análisis Exploratorio de Datos (EDA)

Antes de proceder al entrenamiento, se realizó un **análisis exploratorio** tanto en los datos de entrenamiento como de prueba para comprender su estructura y calidad:

- Se analizaron las variables numéricas y categóricas para identificar su rango, distribución, presencia de sesgos, y valores atípicos (outliers).
- En variables categóricas, se detectaron categorías con baja representación, indicando posibles riesgos de ruido o poca influencia predictiva.
- Se evaluó la proporción de datos faltantes en cada variable y su relación con la variable objetivo.
- Se calculó la correlación entre las variables numéricas y la variable objetivo utilizando matrices de correlación.
- Se evaluó la distribución de la variable objetivo para determinar el balance entre las clases.

3. Depuración y Preprocesamiento de Datos

a. Eliminación de Duplicados

Se aplicaron criterios específicos para garantizar la integridad y singularidad de los datos:

- Cada cliente puede tener varias obligaciones y, a su vez, una obligación puede presentar hasta tres opciones de pago. Este patrón se utilizó como principal criterio para identificar y eliminar duplicados.
- En los casos donde una misma obligación presentaba múltiples valores de *score* o alertas tempranas, se seleccionó el valor más alto como medida para mitigar riesgos asociados al incumplimiento.

b. Reglas de Calidad

Se establecieron procedimientos claros para el tratamiento de valores vacíos o inconsistencias en las variables:

- Variables Numéricas: Valores vacíos, nulos o indicados como None fueron imputados con -1 para facilitar su identificación.
- Variables Alfanuméricas: Datos vacíos o inconsistentes (como N/A o categorías no válidas) también se imputaron con -1.

c. Imputación de Datos Vacíos o Nulos

Para las **variables numéricas** que presentan valores vacíos o nulos, se implementó un enfoque de imputación basado en el **percentil 75** de la distribución de cada variable. Este método garantiza que los datos imputados reflejen una tendencia conservadora hacia valores superiores al promedio, preservando la estructura y características originales de la información.

d. Otras Consideraciones Específicas

- Lote: Se seleccionó el valor más bajo del lote.
- Marca Pago: Se priorizó con base en criterios expertos.

Tras completar estas etapas, las variables numéricas fueron normalizadas utilizando el método StandardScaler, asegurando que los valores estén escalados de manera uniforme.

Ingeniería de Características

Para la selección y optimización de variables, se empleó el método de regularización **Lasso**. Este método:

- Introduce una penalización basada en la norma L1, reduciendo a cero los coeficientes de variables menos significativas.
- Permite identificar y **conservar únicamente** las características más relevantes para el modelo, mejorando así su capacidad predictiva.

Entrenamiento

1. División de Datos

- Los datos se dividieron en dos conjuntos para garantizar una evaluación robusta del modelo:
    - Entrenamiento (70%): Usado para ajustar los parámetros de los modelos.
    - Prueba (30%): Utilizado para evaluar el desempeño del modelo sobre datos no vistos.

2. Modelos Probados

- Se implementaron y evaluaron tres algoritmos de aprendizaje automático:
    1. Regresión Logística: Modelo lineal interpretativo y eficiente.
    2. Decision Tree Classifier: Modelo no lineal basado en reglas de decisión.

3. LightGBM: Modelo basado en gradiente boosting, diseñado para manejar grandes volúmenes de datos y optimizar el tiempo de entrenamiento.

3. Tuneo de Hiperparámetros

- A cada modelo se le aplicó un proceso de **optimización de hiperparámetros** utilizando **GridSearchCV**, que permite probar combinaciones de parámetros de forma sistemática para encontrar la configuración que maximiza el desempeño en un subconjunto de validación.

4. Evaluación de Desempeño

- Los modelos fueron evaluados utilizando el F1-Score como métrica principal.
- Resultados:
    - El modelo LightGBM obtuvo el mejor desempeño en términos de F1-Score, superando a las demás alternativas.

5. Modelo Seleccionado

- Dado su rendimiento superior, el modelo LightGBM fue seleccionado como el modelo ganador para ser utilizado en el proceso de inferencia.

Modelo	F1-SCORE
Regresion Logistica	0.7986
LightGBM	0.8548
Decision Tree Classifier	0.8542

Inferencia

El modelo seleccionado para la inferencia fue determinado en función de su desempeño superior en la métrica de **F1-Score**. A continuación, se describe el proceso de inferencia aplicado:

1. **Selección del Modelo Ganador**
    - Se eligió el modelo con el **F1-Score** más alto durante la etapa de evaluación, garantizando un equilibrio óptimo entre precisión y sensibilidad.
2. **Preparación de la Base de Inferencia**
    - La base de clientes para inferencia (**prueba_op_base_pivot_var_rpta_alt_enmascarado_oot**), suministrada como parte del reto, pasó por un proceso de validación de calidad:

        - Verificación de la inexistencia de valores duplicados a nivel de cliente y obligación, asegurando integridad en los datos.

3. **Ejecución de la Inferencia**
    - Tras completar las validaciones, la base de inferencia fue procesada en **Python** utilizando el modelo ganador.

Productización

El **proceso de productizar un modelo** mediante **MLOps (Machine Learning Operations)** consiste en implementar prácticas y herramientas para llevar un modelo desde su desarrollo hasta su despliegue y operación de manera automatizada y escalable. A continuación, se describe el flujo principal:

1. **Entrenamiento y Validación**
    - El modelo es desarrollado, entrenado y validado a partir de métricas.
    - Se versionan tanto el código como los datos y el modelo generado, utilizando herramientas como los repositorios de Azure Devops.
2. **Empaquetado**
    - El modelo entrenado es empaquetado en un pickle para garantizar su portabilidad y reproducibilidad.
3. **Despliegue Automatizado**
    - Se configura un pipeline de **CI/CD (Integración Continua/Despliegue Continuo)** que permite:
        - Probar automáticamente las actualizaciones del modelo.
        - Desplegar el modelo en entornos de prueba y producción (por ejemplo, como un endpoint en un servicio REST o gRPC en **Docker** o **Kubernetes**).
4. **Integración en Producción**
    - El modelo es alojado en una infraestructura escalable (como AWS SageMaker (AWS), Azure ML o GCP AI Platform).
    - Se asegura la integración con las fuentes de datos en tiempo real o batch y con las aplicaciones que consumirán los resultados.
5. **Monitoreo y Mantenimiento**
    - Se implementan herramientas para monitorear el rendimiento del modelo (latencia, precisión, desviación de datos, etc.).

- El sistema detecta automáticamente cuándo el modelo requiere retraining debido a cambios en los datos (concept drift) y lo actualiza mediante pipelines automatizados.

6. **Automatización de Retraining**
    - Los datos nuevos recolectados se incorporan al pipeline para entrenar y validar nuevas versiones del modelo.
    - Se realiza un despliegue progresivo para validar el nuevo modelo antes de sustituir al anterior (canary testing o A/B testing).

Monitoreo

El monitoreo de los resultados del modelo se llevará a cabo utilizando **Power BI**, una herramienta que permite evaluar su desempeño y generar informes visuales interactivos que agilizan la toma de decisiones. A continuación, se detalla el proceso:

1. Extracción de Resultados

- Los resultados generados por el modelo, que incluyen predicciones, métricas de evaluación y datos reales, se almacenan inicialmente en la **Landing Zone (LZ)**.
- Una métrica clave para supervisar es el **F1-Score**, que permite medir el equilibrio entre precisión y sensibilidad en las predicciones. *Esta es anexada al base resultado del modelo.*

2. Conexión de Power BI

Integración con la Fuente de Datos:

- Power BI se conectará directamente a las tablas ubicadas en la Zona Resultados (ZR), las cuales contienen datos consolidados y transformados provenientes de la LZ.

3. Creación de Informes y Dashboards

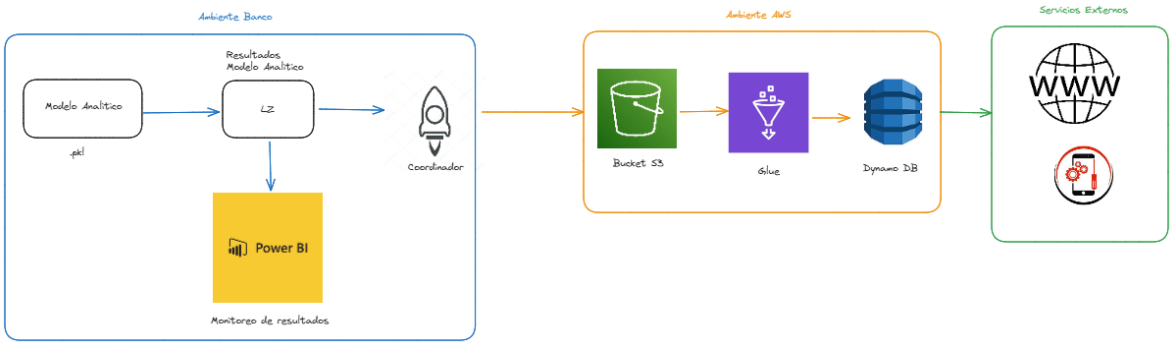
- Mostrar métricas críticas como:
    - La proporción de clientes a quienes se les asignó una opción de pago.
    - Los montos asociados a las opciones de pago asignadas.
- Cruce de datos entre:
    - Clientes que han recibido una opción de pago.
    - Clientes que han aceptado efectivamente dicha opción.

    - Este análisis permitirá calcular cuántos clientes han adoptado la opción de pago ofrecida, proporcionando una métrica de aceptación crucial.

- Gráficos Interactivos:
    - Análisis de tendencias temporales, por ejemplo, el comportamiento del F1-Score en múltiples iteraciones del modelo.
    - Filtros personalizados por región, tipo de cliente o características específicas para un análisis segmentado.

4. **Alertas Automáticas**:
    - Configurar alertas en Power BI para enviar notificaciones cuando las métricas de desempeño caigan por debajo de un umbral crítico.

Arquitectura de Consumo de Información



Nota: Se adjunta imagen para mejor visualización.

El flujo de la información y su consumo se estructura de la siguiente manera, optimizando el proceso desde el análisis de los resultados hasta su disponibilidad para los usuarios finales:

1. **Generación de Resultados por el Modelo Analítico**
    - Una vez que el modelo analítico genera los resultados, estos son almacenados en la **(Landing Zone - LZ)** para su procesamiento inicial.
2. **Proceso de Productización**
    - El proceso de **Productizar** es el encargado de tomar los datos procesados desde la LZ y transferirlos a un **bucket de S3**.
3. **Integración en AWS Glue**
    - Dentro del entorno **AWS**, el servicio **Glue** se utiliza para procesar y transformar la información almacenada en S3.
4. **Disponibilidad en DynamoDB**
    - La información transformada y lista para el consumo se almacena en una tabla de **Amazon DynamoDB**.
    - **DynamoDB** es ideal para este caso debido a su capacidad para gestionar grandes volúmenes de datos con una **baja latencia** en las consultas, garantizando una experiencia rápida y eficiente.
5. **Consumo de la Información**
    - Una vez los datos están en **DynamoDB**, servicios externos pueden consumirlos fácilmente. Ejemplos de estos servicios incluyen:
        - Aplicaciones web.
        - Servicios móviles.

Esta arquitectura garantiza un acceso rápido y fiable a los datos, independientemente del volumen o de las demandas concurrentes.

6. **Monitoreo en del desempeño del modelo en Power BI:**
    - Utilizando la herramienta de Power BI se crea un tablero con visualizaciones que muestran la métrica de desempeño del modelo a lo largo del tiempo. Esto incluye:
        - Tendencias del F1-Score.
        - Comparaciones entre versiones del modelo para identificar mejoras o degradaciones en el desempeño.